JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 07264590
(43)Date of publication of application: 13.10.1995

(51)Int.Cl.
H04N 7/30
H04N 1/41
H04N 5/92

(21)Application number: 06056200
(22)Date of filing: 25.03.1994

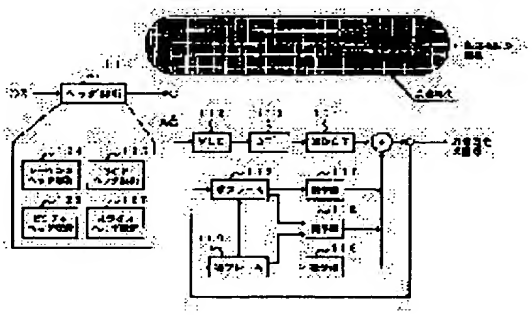
(71)Applicant: NEC CORP
(72)Inventor: UEDA HIROAKI

(54) IMAGE REGENERATING DEVICE

(57)Abstract:

PURPOSE: To quickly regenerate a code based upon discrete cosine transformation by simple constitution without lossing picture quality by providing this picture regenerating device with a means for analyzing a code and judging whether the code fixes a parameter or not.

CONSTITUTION: In the image regenerating device, a header analysis part 111 analyzes a read code, judges whether the code fixes a parameter or not, and when the code fixes its parameter, high speed reproducing is executed. In the case of high speed regeneration, only an I picture to be an intra-frame code is regarded as a slice width fixed escape code and a highly efficiently compressed code is decoded by a VLD2 part. The decoded data are reversely quantized by a Q2-1 part and reverse DCT is applied to the reversely quantized data by a reverse DCT2 part to display an enlarged picture. When a parameter is not



fixed, normal regeneration is executed through a VLD part 112. Consequently high speed processing can be executed in accordance with the processing capacity of a regenerating device.

LEGAL STATUS

[Date of request for examination]	15.11.1995
[Date of sending the examiner's decision of rejection]	11.08.1999
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	1114617
[Date of requesting appeal against examiner's decision of rejection]	10.09.1999
[Date of extinction of right]	

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平7－264590

(43)公開日　平成７年(1995)10月13日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｈ０４Ｎ　7/30				
1/41	B			
5/92				
			Ｈ０４Ｎ　7/ 133	
			5/ 92　　Ｈ	

審査請求　未請求　請求項の数６　ＯＬ　（全 11 頁）

(21)出願番号　特願平6－56200

(22)出願日　平成６年(1994)３月25日

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号

(72)発明者　上田　裕明
東京都港区芝五丁目７番１号　日本電気株式会社内

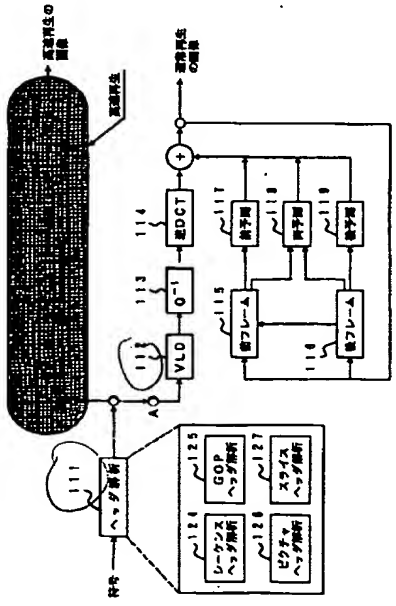
(74)代理人　弁理士　後藤　洋介　（外２名）

(54)【発明の名称】　画像再生装置

(57)【要約】

【目的】　ＤＣＴをベースとした画像符号化方式において再生機の能力に応じて画質をできるだけ損なわずに、簡単な構成によって高速に再生する画像再生装置を提供すること。

【構成】　画像を小ブロックに分割して、各ブロック毎に離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム内符号と、画像を小ブロックに分割して、各ブロック毎に現フレームとその前後のフレームで最も差分が小さくなるようなブロックを検索して動き補償を行い、現フレームのブロックと動き補償されたフレームのブロックで差分を取り、その差分ブロックに離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム間符号で圧縮符号化された動画像を再生する画像再生装置において、符号を解析してパラメータを固定した符号であるかどうか判定する手段１１１を備えている。



1

【特許請求の範囲】

【請求項1】　画像を小ブロックに分割して、各ブロック毎に離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム内符号と、画像を小ブロックに分割して、各ブロック毎に現フレームとその前後のフレームで最も差分が小さくなるようなブロックを検索して動き補償を行い、現フレームのブロックと動き補償されたフレームのブロックで差分を取り、その差分ブロックに離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム間符号で圧縮符号化された動画像を再生する画像再生装置において、符号を解析してパラメータを固定した符号であるかどうか判定する手段を備えたことを特徴とする画像再生装置。

【請求項2】　請求項1に記載した画像再生装置において、更に、表示する画像を拡大する手段を有することを特徴とする画像再生装置。

【請求項3】　請求項1に記載した画像再生装置において、更に、フレーム内符号のみに固定して伸張処理を行う手段を有することを特徴とする画像再生装置。

【請求項4】　請求項1に記載した画像再生装置において、更に、スライス幅を固定値に固定して伸張処理を行う手段を有することを特徴とする画像再生装置。

【請求項5】　請求項1に記載した画像再生装置において、更に、高周波成分を0と見なして逆離散コサイン変換の算出を行う手段を有することを特徴とする画像再生装置。

【請求項6】　請求項1に記載した画像再生装置において、更に、固定長の符号のみに固定して復号する手段を有することを特徴とする画像再生装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、離散コサイン変換（ＤＣＴ、以下ＤＣＴという）をベースとした符号化方式（ＪＰＥＧやＭＰＥＧなど）で圧縮符号化された画像符号を再生する装置に関する。

【0002】

【従来の技術】画像をデジタル化してＣＤ－ＲＯＭやハードディスクなどの記録媒体に記録する場合、そのデータ量は巨大なものとなるため通常は圧縮符号化して記録される。

【0003】圧縮符号化方式の中で使用されることが多いＤＣＴをベースとした符号化方式があるが、これはＪＰＥＧ（Ｊｏｉｎｔ　Ｐｈｏｔｏｇｒａｐｈｉｃ　Ｅｘｐｅｒｔ　Ｇｒｏｕｐ）やＭＰＥＧ（Ｍｏｖｉｎｇ　Ｐｉｃｔｕｒｅｓ　Ｅｘｐｅｒｔ　Ｇｒｏｕｐ）などの国際標準である符号化方式で採用されている。

【0004】従来のＤＣＴをベースとした符号化方式による画像符号の再生についてＭＰＥＧを例として図面を参照して説明する。図16はＭＰＥＧに準拠した画像符号の再生を説明する図である。図16に示すように符号

2

を読み込んで、ヘッダ解析201で符号の種別などを解析する。ＭＰＥＧではフレーム内符号であるＩピクチャと、前方向のみのフレーム間符号であるＰピクチャと、前後の双方向のフレーム間符号であるＢピクチャの3種類に分かれている。

【0005】Ｉピクチャの場合はＶＬＤ202で高能率圧縮された可変長のハフマン符号を復号して、Q^{-1}203で逆量子化して、ＩＤＣＴ204で逆ＤＣＴ処理によりブロックの画素の値を算出して、画像を伸張する。

【0006】また、Ｐピクチャの場合はＶＬＤ202で復号して、Q^{-1}203で逆量子化して、ＩＤＣＴ204で逆ＤＣＴ処理によりブロックの差分値を算出して、前予測207により前フレーム205に格納された前フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

【0007】また、Ｂピクチャの場合はＶＬＤ202で復号して、Q^{-1}203で逆量子化して、ＩＤＣＴ204で逆ＤＣＴ処理によりブロックの差分値を算出して、両予測208または後予測209により前フレーム205に格納された前フレームの動き補償したブロックと後フレーム206に格納された後フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

【0008】このように国際標準であるＭＰＥＧに基づいた再生機であればどの装置でもＭＰＥＧの符号を再生することができる。しかし、逆ＤＣＴやＶＬＤなどの処理にはＣＰＵの負担が大きいので、高速なＣＰＵでなければ高速再生はできない。例えば、ＪＰＥＧやＭＰＥＧなどの再生機で15フレーム／秒の処理を行うには1フレームの再生処理は約66ミリ秒で行う必要がある。もし、ハフマン符号を復号するのに30ミリ秒かかり、逆量子化するのに10ミリ秒かかり、逆ＤＣＴに20ミリ秒かかり、表示に20ミリ秒かかると全体の処理時間が80ミリ秒かかるので、1フレームの画像を再生するのに14ミリ秒遅れる。

【0009】そこで、ＣＰＵの負担を少なくするために符号化方式の一部を変えてＤＣＴ計算の代わりに近傍画素との差分計算にしたり、量子化を省いたり、可変長の符号の代わりに4ビット単位の符号で符号化することなどによって、低速なＣＰＵでも高速に再生できるようにすることが考えられる。

【0010】この従来例として特開昭63－95791があるが、この方式ではブロックのデータの長さが等しくなるように可変長データを分配することにより、ＣＰＵに負担をかけずに再生できるようにしている。

【0011】また、特開平4－56492では1ブロックのＤＣＴ変換後の係数が全て0である場合には無効ブロックとして、無効ブロックを示す情報のみを別信号で送ることにより、処理量を削減している。

【0012】また、ＣＰＵの負担を少なくするために、符号化するときに圧縮符号量を制御する方法も考えられ

る。その従来例として、特開平4－329089があるが、この方式ではフレーム内で小領域に分割されたブロック毎の情報量に応じて符号量制御を行い、各ブロック毎に割り当てる符号量とステップサイズを最適な状態に設定することで、符号量を制御している。

【0013】

【発明が解決しようとする課題】しかしながら上記のような符号化方式の一部を変える画像圧縮では国際標準の符号化方式と互換性が取れなくなるために、専用の再生機が必要となる問題点がある。

【0014】また、圧縮符号量の制御を行う画像圧縮では複数回の圧縮を繰り返す必要があるため、圧縮符号を作成するのに時間がかかる。また、再生機の処理能力に応じた符号量制御ではないため、処理速度の遅い再生機ではリアルタイムの再生ができない。また、圧縮符号量が決っているので高速な再生機でも決められた画質の符号しか再生できないという問題点がある。

【0015】そこで、本発明の目的は再生機の能力に応じて画質をできるだけ損なわずに、簡単な構成によって、DCTをベースとした符号を高速に再生できるようにした画像再生装置や画像再生方式を提供することにある。

【0016】

【課題を解決するための手段】本発明によれば、画像を小ブロックに分割して、各ブロック毎に離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム内符号と、画像を小ブロックに分割して、各ブロック毎に現フレームとその前後のフレームで最も差分が小さくなるようなブロックを検索して動き補償を行い、現フレームのブロックと動き補償されたフレームのブロックで差分を取り、その差分ブロックに離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム間符号で圧縮符号化された動画像を再生する画像再生装置において、符号を解析してパラメータを固定した符号であるかどうか判定する手段を備えたことを特徴とする画像再生装置が得られる。

【0017】また、本発明によれば、更に、表示する画像を拡大する手段を有することを特徴とする画像再生装置が得られる。

【0018】また、本発明によれば、更に、フレーム内符号のみに固定して伸張処理を行う手段を有することを特徴とする画像再生装置が得られる。

【0019】また、本発明によれば、更に、スライス幅を固定値に固定して伸張処理を行う手段を有することを特徴とする画像再生装置が得られる。

【0020】また、本発明によれば、更に、高周波成分を0と見なして逆離散コサイン変換の算出を行う手段を有することを特徴とする画像再生装置が得られる。

【0021】また、本発明によれば、更に、固定長の符号のみに固定して復号する手段を有することを特徴とする画像再生装置が得られる。

【0022】更に、本発明によれば、上記各手段を組み合わせた画像再生装置を得ることもできる。

【0023】

【作用】本発明によれば、再生機の処理能力に応じて画質をできるだけ損なわずに、高速に再生できる。また、国際標準である符号化方式と互換性が取れるので、専用の再生機を必要としない。

【0024】

【実施例】次に本発明についてMPEGを例として図面を参照して説明する。図1は本発明の一実施例を示す画像再生装置のブロック図である。図1の画像再生装置はヘッダ解析111とVLD112とQ^{-1}113と逆DCT114と前フレーム115と後フレーム116と前予測117と両予測118と後予測119から構成される。また、ヘッダ解析111はシーケンスヘッダ解析124とGOPヘッダ解析125とピクチャヘッダ解析126とスライスヘッダ解析127から構成される。画像再生装置は符号を読み込んでヘッダ解析111を行う。その時、パラメータを固定した符号であるかどうか判断して、そうである場合は高速再生を行う。高速再生の場合にはIピクチャのみ、スライス幅固定、エスケープ符号のみと見なして、VLD2（120）で高能率圧縮された符号を復号して、$Q2^{-1}$121で逆量子化して、逆DCT2（122）で逆DCTを行って、拡大123で再生する画像を拡大して表示する。高速再生でない場合には、フレーム内符号であるIピクチャと、前方向のみのフレーム間符号であるPピクチャと、前後の双方向のフレーム間符号であるBピクチャの3種類の符号を伸張する。

【0025】Iピクチャの場合はVLD112で復号して、Q^{-1}113で逆量子化して、逆DCT114で逆DCT処理によりブロックの画素の値を算出して、画像を伸張する。

【0026】また、Pピクチャの場合はVLD112で復号して、Q^{-1}113で逆量子化して、逆DCT114で逆DCT処理によりブロックの差分値を算出して、前予測117により前フレーム115に格納された前フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

【0027】また、Bピクチャの場合はVLD112で復号して、Q^{-1}113で逆量子化して、逆DCT114で逆DCT処理によりブロックの差分値を算出して、両予測118または後予測119により前フレーム115に格納された前フレームの動き補償したブロックと後フレーム116に格納された後フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

【0028】また、図2にMPEGに準拠した符号フォーマットの階層図を示す。MPEGの符号は図2に示すようにいくつかの階層構造となっている。一番上の階層

がビデオ・シーケンスであり、複数のＧＯＰ（Ｇｒｏｕｐ　Ｏｆ　Ｐｉｃｔｕｒｅ）から構成される。ＧＯＰは複数のピクチャから構成され、１つのピクチャが１枚の画像を示している。ピクチャは任意の領域に分割された複数のスライスから構成される。スライスは左から右へ、または上から下への順序で並んだ複数のマクロブロックから構成される。マイクロブロックは１６×１６ドットのブロックを更に８×８ドットのブロックに分割した輝度成分（Ｙ１，Ｙ２，Ｙ３，Ｙ４）と輝度成分に一致する領域の８×８ドットのブロックの色差成分（Ｃｂ，Ｃｒ）の６個のブロックから構成される。８×８ドットのブロックが符号化の最小単位となる。

【００２９】また、図３にＭＰＥＧに準拠した符号フォーマットの構成図を示す。ＭＰＥＧの符号は図３に示すように各階層毎に（１）シーケンスヘッダと、（２）ＧＯＰ（Ｇｒｏｕｐ　Ｏｆ　Ｐｉｃｔｕｒｅ）ヘッダと、（３）ピクチャヘッダと、（４）スライスヘッダと、（５）マクロブロックヘッダと、（６）ブロックの符号とから構成される。

【００３０】シーケンスヘッダにはＵｓｅｒＤａｔａ（１１）に示すようにユーザが自由に定義できるユーザデータの領域があり、その領域にパラメータを固定したことを示すフラグを格納する。そのユーザデータ（１１）は固定パラメータで圧縮した符号であることを示す識別子“Ｃｕｓｔｏｍ”（１２）と、固定パラメータのフラグ（１３）で構成される。固定パラメータのフラグ（１３）は②伸張時に拡大することを示すビットと、③Ｉピクチャのみに固定することを示すビットと、④スライス幅はピクチャサイズに固定することを示すビットと、⑤高周波成分をカットすることを示すビットと、⑥エスケープ符号のみに固定することを示すビットで構成される。

【００３１】次に、以上の構成を有する本実施例の画像再生装置における画像再生処理について説明する。

【００３２】図４は画像再生装置の拡大を説明する図である。図４では２×２画素の画像を縦２倍、横２倍に拡大する場合の例を示している。図４の（Ａ）は２×２画素の各画素の値を示し、（Ｃ）は縦２倍、横２倍に拡大後の４×４画素の各画素の値を示している。また、図４の（Ｂ）は１画素を４つの画素に変換する処理を示している。図４の例では（Ａ）から１画素取り出して、それを（Ｂ）のように４つの画素に変換して、縦２倍、横２倍に拡大した画像（Ｃ）に書き込んでいる。例えば、（Ａ）の右上の画素（２９）は（Ｂ）の変換により、Ｐ１（２９）とＰ２（２９）とＰ３（２９）とＰ４（２９）となり、（Ｃ）の対応する領域に書き込まれる。

【００３３】図４の例では縦２倍、横２倍に拡大するときに画質を向上するために画素の値の補正（ディサ処理）を行っていないが、ディサ処理を行っても良い。この場合例えば、図４の（Ｂ）でＰ１＝Ｐ０＋ａ、Ｐ２＝Ｐ０＋ｂ、Ｐ３＝Ｐ０＋ｃ、Ｐ４＝Ｐ０＋ｄ（ａ，ｂ，ｃ，ｄは任意の整数）として、４倍拡大を行う。また、倍率は縦２倍、横２倍としているが、３以上の整数倍でも良い。

【００３４】また、図５は図４の動作を説明するフローチャートである。図５に示すように拡大は拡大前の画像から画素を１つ読み込んで（ステップ１）、読み込んだ１つの画素（Ｐ０）を４つの画素（Ｐ１＝Ｐ０、Ｐ２＝Ｐ０、Ｐ３＝Ｐ０、Ｐ４＝Ｐ０）に変換して（ステップ２）、変換した４つの画素を拡大後の画像に書き込む（ステップ３）。次に全画素を処理したかどうか判断して（ステップ４）、そうでない場合はステップ１に戻り、そうでない場合は処理を終了する。

【００３５】このように伸張した画像を拡大して表示することにより、伸張時間を短縮でききる。例えば、縦２倍、横２倍に拡大する場合は伸張時間は１／４に短縮できる。

【００３６】また、拡大の大部分の処理はメモリのリード／ライトである。伸張の処理にはメモリのリード／ライトだけでなく、逆ＤＣＴや逆量子化での乗算や可変長符号の復号でのビットシフトの処理がある。メモリのリード／ライトの方が乗算やビットシフトよりもＣＰＵの負担が小さいので、拡大処理を含めても、拡大しない場合よりも拡大した場合の方が高速に処理できる。

【００３７】図６はピクチャ構成を固定にした場合の処理を説明する図である。ＭＰＥＧではＩピクチャ（フレーム内符号）とＰピクチャ（前方向予測符号）とＢピクチャ（双方向予測符号）の３つのピクチャがあるが、図６の（Ａ）はＩピクチャとＰピクチャとＢピクチャから成るピクチャ構成を示し、（Ｂ）はＩピクチャのみに固定したピクチャ構成を示している。Ｐピクチャの場合は前ピクチャの画像を参照して前予測を行う必要があり、またＢピクチャの場合は前フレームの画像や後フレームの画像を参照して後予測や両予測を行う必要があるので、Ｉピクチャの場合と比べて処理時間がかかる。図６の（Ａ）の例では１フレーム目はＩピクチャであり参照フレームは無い。２フレーム目と２フレーム目はＢピクチャであり、前フレームである１フレーム目の画像と後フレームである４フレーム目の画像を参照して、後予測や両予測を行う。４フレームはＰピクチャであり、前フレームである１フレーム目の画像を参照して前予測を行う。図６の（Ｂ）の例では１フレーム目から４フレーム目まで全てＩピクチャなので、前予測や後予測や両予測を行う必要は無い。

【００３８】また、図７は図６の動作を説明するフローチャートである。図７の（Ａ）はＩピクチャとＰピクチャとＢピクチャから成るピクチャ構成を処理する場合を示し、（Ｂ）はＩピクチャのみに固定したピクチャ構成を処理する場合を示し、（Ｃ）は（Ａ）と（Ｂ）のどちらを処理するか判断する処理を示している。図７の

（A）に示すピクチャ処理１はピクチャヘッダからピクチャの種別を読み込んで（ステップ１０）、ピクチャの種別を判断して（ステップ１１）、Ｉピクチャの場合はＩピクチャの処理を行う（ステップ１２）。Ｐピクチャの場合はＰピクチャの処理を行う（ステップ１３）。Ｂピクチャの場合はＢピクチャの処理を行う（ステップ１３）。図7の（B）に示すピクチャ処理２はＩピクチャの処理を行う（ステップ１５）。図7の（C）に示すピクチャヘッダ解析はパラメータ固定であるかどうか判断して（ステップ１６）、そうでない場合はピクチャ処理１を行う（ステップ１７）。そうである場合はピクチャ処理２を行う（ステップ１８）。

【００３９】このようにピクチャ構成をＩピクチャのみに固定することにより、ピクチャヘッダからピクチャ種別を読み込んで判断する処理を省略できて、Ｐピクチャの処理の前予測やＢピクチャの処理の後予測や両予測を省略できるので、高速に処理できる。

【００４０】図８はスライス幅を固定した場合の処理を説明する図である。ＭＰＥＧでは１フレームの画像を１６×１６画素を単位としていくつかの領域に分割している。分割した領域の符号の先頭にはスライスヘッダが挿入されて、画像のどの領域であるかを示している。図８の（A）はスライス幅を固定していない場合を示し、図８の（B）はスライス幅をピクチャサイズに固定した場合を示している。図８の（A）は画像を５つの領域（１～５）に分割している。１から５までの各領域の符号の先頭にはスライスヘッダが挿入される。図８の（B）ではスライス幅をピクチャサイズに固定しているので、領域を分割していない。そのため、スライスヘッダも１つしかない。

【００４１】また、図９は図８の動作を説明するフローチャートである。図９の（A）はスライス幅を固定していない場合の処理を示し、（B）はスライス幅をピクチャサイズに固定した場合の処理を示し、（C）は（A）と（B）のどちらを処理するか判断する処理を示している。図９の（A）に示すスライス処理１はスライスヘッダを読み込んで（ステップ２０）、マクロブロックの処理を行い（ステップ２１）、次のヘッダがスライスヘッダであるかどうか判断して（ステップ２２）、そうである場合はステップ２０へ戻る。そうでない場合は全マクロブロックの処理が終了したかどうか判断して（ステップ２３）、そうでない場合はステップ２１へ戻り、そうである場合は処理を終了する。図９の（B）に示すスライス処理２はスライスヘッダを読み込んで（ステップ２４）、マクロブロックの処理を行い（ステップ２５）、全マクロブロックの処理が終了したかどうか判断して（ステップ２６）、そうでない場合はステップ２４へ戻り、そうである場合は処理を終了する。図９の（C）に示すスライスヘッダ解析はパラメータ固定であるかどうか判断して（ステップ２７）、そうでない場合はスライス処理１を行う（ステップ２８）。そうである場合はスライス処理２を行う（ステップ２９）。

【００４２】このようにスライス幅をピクチャサイズに固定することにより、スライスヘッダを読み込んで解析する処理回数を削減できるので、高速に処理できる。

【００４３】図１０は高周波成分を０と見なして逆ＤＣＴを行う場合の処理を説明する図である。図１０の（A）は高周波成分を０と見なさない場合を示し、図１０の（B）は高周波成分を０と見なす場合を示している。図１０の（A）は８×８ブロックの周波数成分の低周波から高周波へのスキャン順序（ジグザグスキャン）を示している。図１０の（B）もジグザグスキャンを示しているが、２２番目以降の高周波成分を全て０と見なしている。自然画像の場合は画像情報が低周波数成分に集中するので、高周波数成分を０と見なしても、画質は大きく劣化して見えない。

【００４４】また、図１１は高周波成分を０と見なさずに逆ＤＣＴを行う場合の動作を説明するフローチャートである。図１１に示す逆ＤＣＴは変数ｙに０を格納して（ステップ３０）、変数ｖに０を格納して（ステップ３１）、変数ｄｄに０を格納して（ステップ３２）、変数ｕに０を格納して（ステップ３３）。次に変数ｄｄに逆ＤＣＴを行う８×８ブロックのバッファのＢｕｆｆｅｒ（ｙ，ｕ）と逆ＤＣＴの係数であるｃｏｓ（（２ｖ＋１）ｕπ）／２の積を加算する（ステップ３４）。次に変数ｕに１を加算して（ステップ３５）、変数ｕの値が８より小さいかどうか判断して（ステップ３６）、そうである場合はステップ３４へ戻る。そうでない場合は一時的な格納バッファのｔ（ｖ，ｙ）に変数ｄｄの値を格納する（ステップ３７）。次に変数ｖに１を加算して（ステップ３８）、変数ｖの値が８より小さいかどうか判断して（ステップ３９）、そうである場合はステップ３２へ戻る。次に変数ｙに１を加算して（ステップ４０）、変数ｙの値が８より小さいかどうか判断して（ステップ４１）、そうである場合はステップ３１へ戻る。そうでない場合は変数ｙに０を格納する（ステップ４２）。次に変数ｘに０を格納して（ステップ４３）、変数ｄｄに０を格納して（ステップ４４）、変数ｕに０を格納する（ステップ４５）。次に変数ｄｄにｔ（ｙ，ｕ）とｃｏｓ（（２ｘ＋１）ｕπ）／２の積を加算する（ステップ４６）。次に変数ｕに１を加算して（ステップ４７）、変数ｕの値が８より小さいかどうか判断して（ステップ４８）、そうである場合はステップ４６へ戻る。そうでない場合はＢｕｆｆｅｒ（ｘ，ｙ）に変数ｄｄの値を格納する（ステップ４９）、次に変数ｘに１を加算して（ステップ５０）、変数ｘの値が８より小さいかどうか判断して（ステップ５１）、そうである場合はステップ４４へ戻る。次に変数ｙに１を加算して（ステップ５２）、変数ｙの値が８より小さいかどうか判断して（ステップ５３）、そうである場合はステップ４４へ

戻る。そうでない場合は処理を終了する。

【００４５】また、図１２は高周波成分を０と見なして逆ＤＣＴを行う場合の動作を説明するフローチャートである。図１２は２２番目以降の周波数成分を０と見なした例である。図１２に示す逆ＤＣＴ２は変数ｙに０を格納して（ステップ６０）。変数ｖに０を格納して（ステップ６１）、変数ｄｄに０を格納して（ステップ６２）、変数ｕに０を格納する（ステップ６３）。次に変数ｄｄに逆ＤＣＴを行う８×８ブロックのバッファのＢｕｆｆｅｒ（ｙ，ｕ）と逆ＤＣＴの係数であるｃｏｓ（（２ｖ＋１）ｕπ）／２の積を加算する（ステップ６４）。次に変数ｕに１を加算して（ステップ６５）、変数ｕの値がｕｍａｘ（ｙ）の値より小さいかどうか判断して（ステップ６６）、そうである場合はステップ６４へ戻る。そうでない場合は一時的な格納バッファのｔ（ｖ，ｙ）に変数ｄｄの値を格納する（ステップ６７）。次に変数ｖに１を加算して（ステップ６８）、変数ｖの値が８より小さいかどうか判断して（ステップ６９）、そうである場合はステップ６２へ戻る。次に変数ｙに１を加算して（ステップ７０）、変数ｙの値が６より小さいかどうか判断して（ステップ７１）、そうである場合はステップ６１へ戻る。そうでない場合は変数ｙに０を格納する（ステップ７２）。次に変数ｘに０を格納して（ステップ７３）、変数ｄｄに０を格納して（ステップ７４）、変数ｕに０を格納する（ステップ７５）。次に変数ｄｄにｔ（ｙ，ｕ）とｃｏｓ（（２ｘ＋１）ｕπ）／２の積を加算する（ステップ７６）。次に変数ｕに１を加算して（ステップ７７）、変数ｕの値が６より小さいかどうか判断して（ステップ７８）、そうである場合はステップ７６へ戻る。そうでない場合はＢｕｆｆｅｒ（ｘ，ｙ）に変数ｄｄの値を格納する（ステップ７９）。次に変数ｘに１を加算して（ステップ８０）、変数ｘの値が８より小さいかどうか判断して（ステップ８１）、そうである場合はステップ７４へ戻る。次に変数ｙに１を加算して（ステップ８２）、変数ｙの値が８より小さいかどうか判断して（ステップ８３）、そうである場合はステップ７４へ戻る。そうでない場合は処理を終了する。

【００４６】図１２の例では２２番目以降を０と見なしているが、再生機の処理能力に応じて１～６４の値でも良い。

【００４７】このように高周波成分を０と見なさない場合の乗算回数は８×８×８×２＝１０２４回であり、２２番目以降の周波数成分を０と見なした場合の乗算回数は８×（６＋５＋４＋３＋２＋１）×６＋６×８×８＝５１０回であるので、高周波成分を０と見なして逆ＤＣＴを行った方が高速に処理できる。

【００４８】図１３はエスケープ符号の構成図である。ＭＰＥＧには固定ピッチ長のエスケープ符号があり、それは（１）エスケープ符号であることを示す符号（０００１）と（２）ラン長（無効係数の数）と（３）レベル（有効係数の値：－１２８～＋１２８）から構成される。

【００４９】図１４は通常の符号とエスケープ符号の例を示した表である。図１４に示すように通常の符号は２～１７ビットの可変長の符号であるが、エスケープ符号は全て２０ビットの固定長の符号となる。

【００５０】また、図１５はＶＬＤの動作を説明するフローチャートである。図１５の（Ａ）は通常の符号の場合の処理を示し、図１５の（Ｂ）はエスケープ符号のみに固定した場合の処理を示している。図１５の（Ａ）に示すＶＬＤは符号を８ビット取り出し（ステップ９０）、８ビットの値から符号の種類を判断して（ステップ９１）、８ビット以下の符号であるかどうかを判断して（ステップ９２）、そうでない場合はステップ９４へ進む。そうでない場合はさらに必要なビット数分取り出して（ステップ９３）、符号のビット数を符号バッファのポインタに加算して（ステップ９４）、符号に対応するラン長とレベルを取り出して（ステップ９５）、ラン長とレベルを返す（ステップ９６）。図１５の（Ｂ）に示すＶＬＤ２は符号を８ビット取り出し（ステップ９７）、ｂｉｔ０～ｂｉｔ７からレベルを取り出し（ステップ９８）、ｂｉｔ８～ｂｉｔ１３からラン長を取り出し（ステップ９９）、２０ビットを符号バッファのポインタに加算して（ステップ１００）、ラン長とレベルを返す（ステップ１０１）。

【００５１】このように通常の符号の場合は符号ビット数を調べて、その値に応じて処理しなければならないが、エスケープ符号のみの場合は２０ビットの固定長なので、符号ビット数を調べる必要がなく、高速に処理できる。

【００５２】

【発明の効果】以上説明したように本発明は、パラメータを固定して伸張することができるので、再生機の処理野兎力に応じて画質をできるだけ損わずに、高速に再生できる。また、国際標準である符号化方式と互換性が取れるので、専用の再生機を必要としない。

【図面の簡単な説明】

【図１】本発明の一実施例の動画像再生装置のフロー図である。

【図２】ＭＰＥＧの符号の階層図である。

【図３】ＭＰＥＧの符号の構成図である。

【図４】本発明の拡大処理を説明する図である。

【図５】本発明の拡大処理の動作を示すフローチャートである。

【図６】本発明のピクチャ構成を固定した処理を説明する図である。

【図７】本発明のピクチャ処理の動作を示すフローチャートである。

【図８】本発明のスライス幅を固定した処理を説明する

11

図である。

【図９】本発明のスライス処理の動作を示すフローチャートである。

【図１０】本発明の高周波成分を０と見なす処理を説明する図である。

【図１１】逆ＤＣＴの動作を示すフローチャートである。

【図１２】本発明の逆ＤＣＴの動作を示すフローチャートである。

【図１３】エスケープ符号の構成図である。

【図１４】通常の符号とエスケープ符号の例を示した表である。

【図１５】本発明のＶＬＤの動作を示すフローチャートである。

【図１６】従来例の動画再生装置のフロー図である。

【符号の説明】

１１１　　ヘッダ解析

12

１１２　　ＶＬＤ

１１３　　Q^{-1}

１１４　　逆ＤＣＴ

１１５　　前フレーム

１１６　　後フレーム

１１７　　前予測

１１８　　両予測

１１９　　後予測

１２０　　ＶＬＤ２

１２１　　$Q2^{-1}$

１２２　　逆ＤＣＴ２

１２３　　拡大

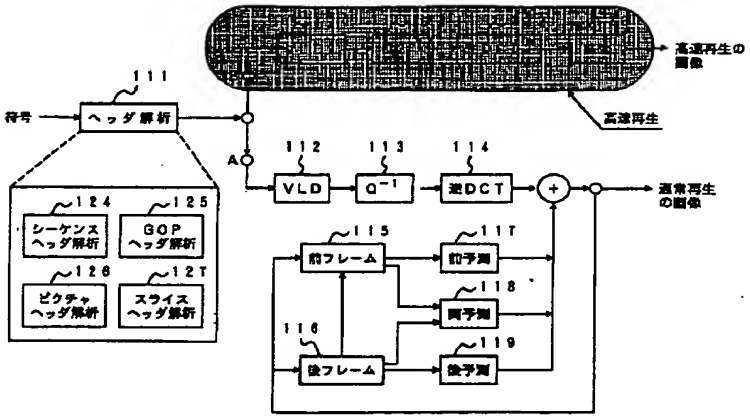
１２４　　シーケンスヘッダ解析

１２５　　ＧＯＰヘッダ解析

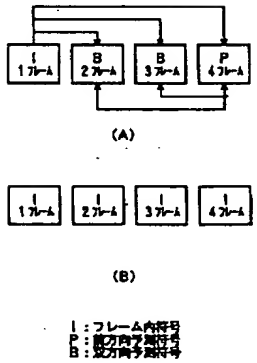
１２６　　ピクチャヘッダ解析

１２７　　スライスヘッダ解析

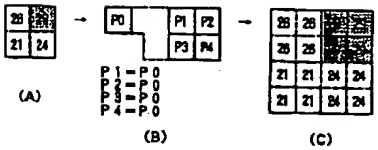
【図１】



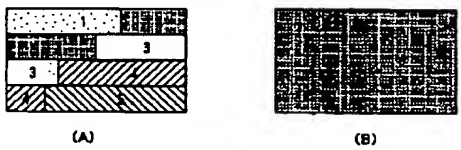
【図６】



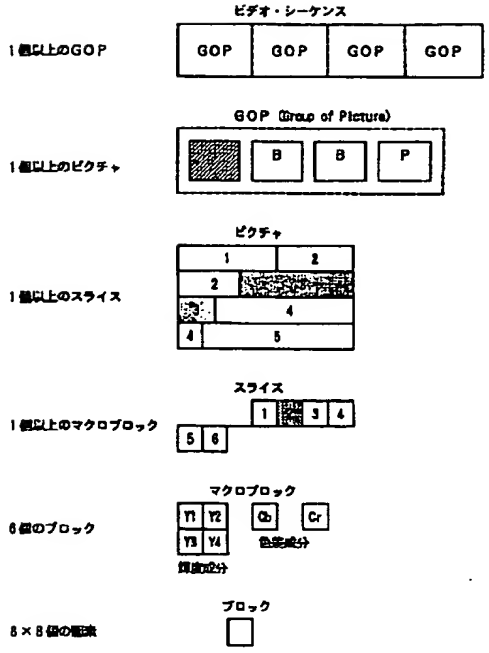
【図４】



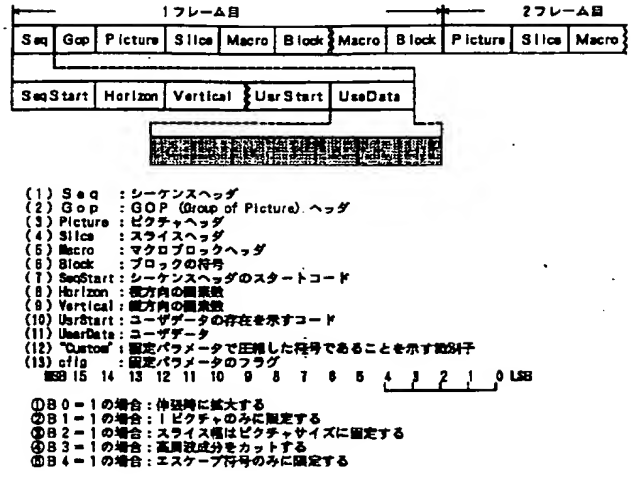
【図８】



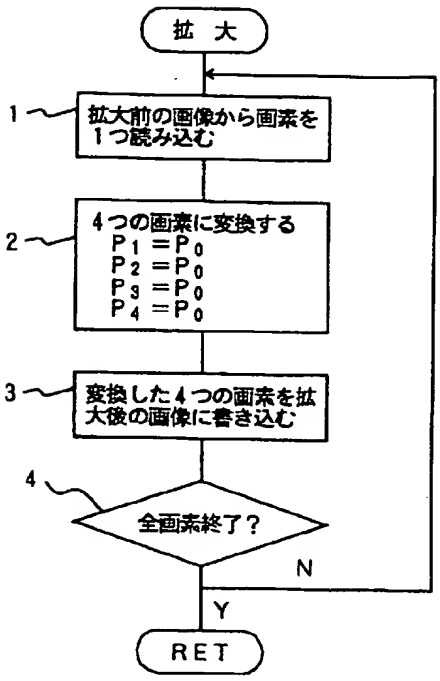
【図2】



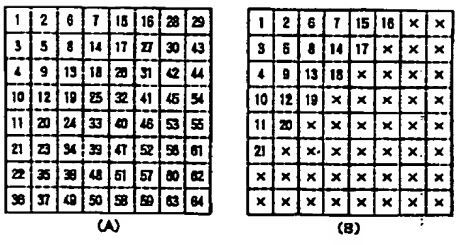
【図3】



【図5】



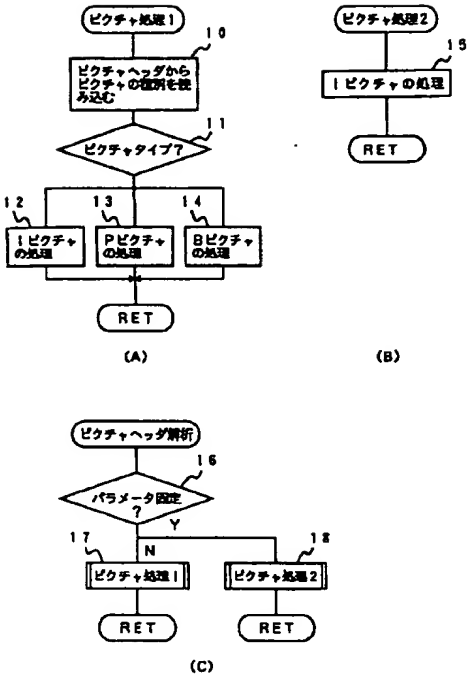
【図10】



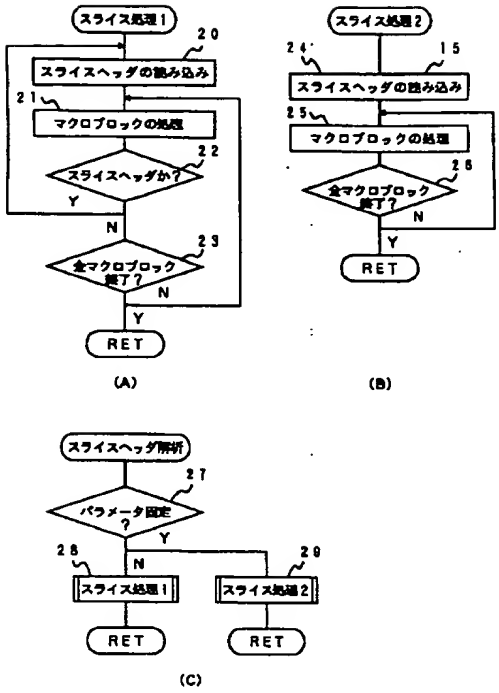
【図14】

ラン	レベル	通常符号 2～17ビット	エスケープ符号 20ビット固定
2	1	0 1010	0000 0100 0010 0000 0001
1	4	000 0001 1000	0000 0100 0100 0000 0100
2	5	00 0000 0010 1000	0000 0100 0101 0000 0101

【図7】



【図9】

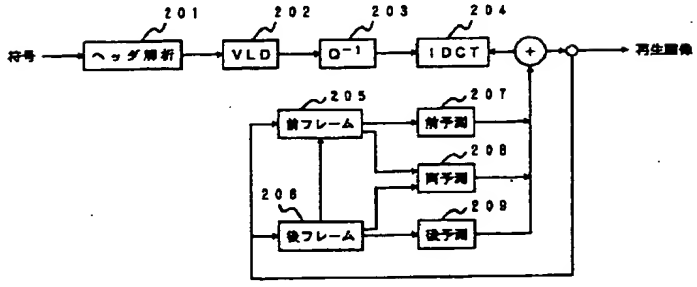


【図13】

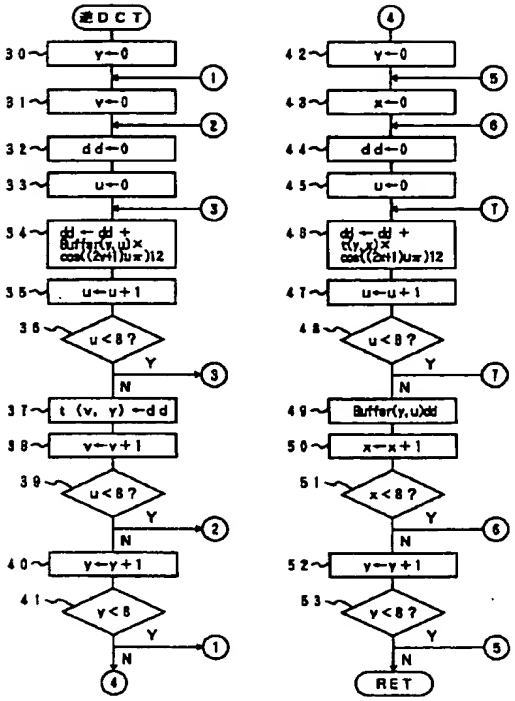
← Esc →						← Run →						← LeveL →							
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x

(1) Esc　：エスケープ符号であることを示す符号
(2) Run　：ラン長（実数値域の値）
(3) Level：レベル（有効値域の値：－128～＋128）

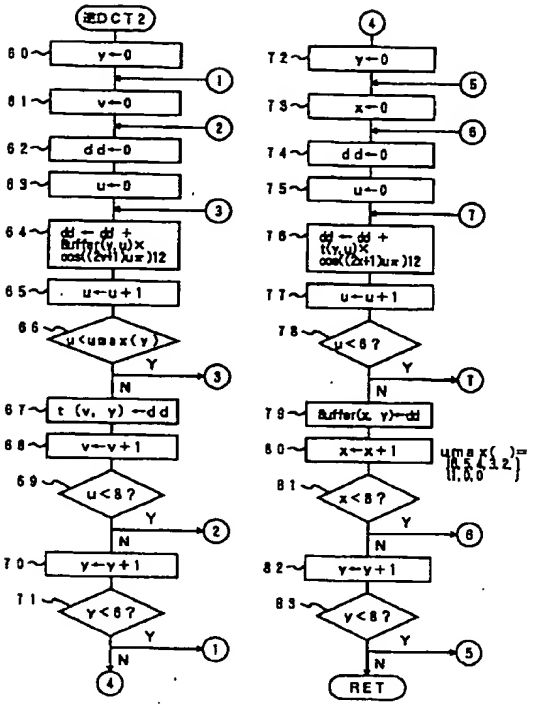
【図16】



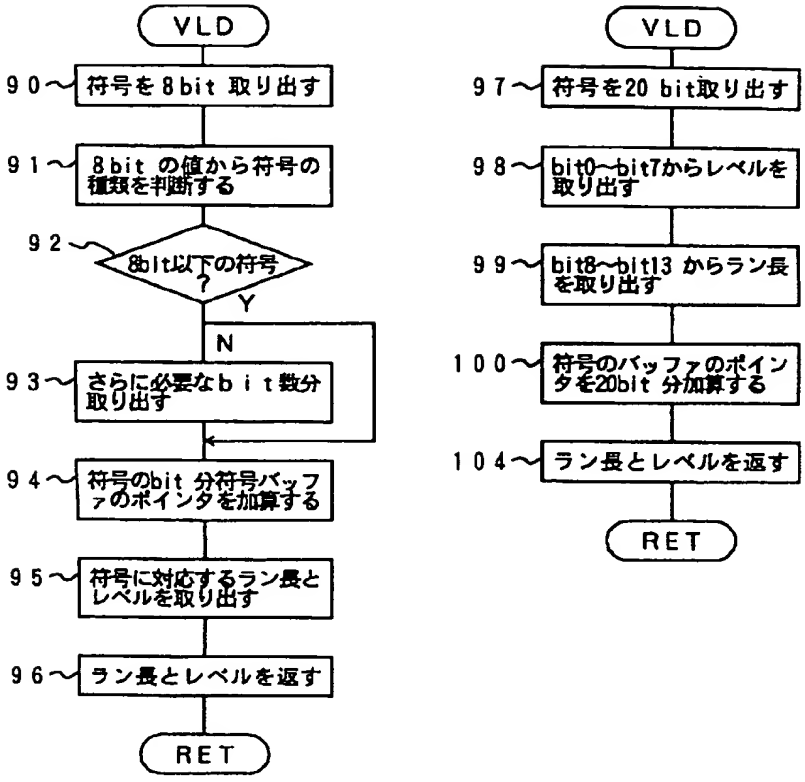
【図１１】


逆ＤＣＴ
３０ y←0
３１ v←0
３２ dd←0
３３ u←0
３４ dd←dd+Buffer(y,u)×cos((2x+1)uπ/16)
３５ u←u+1
３６ u<8?
３７ t(v,y)←dd
３８ v←v+1
３９ v<8?
４０ y←y+1
４１ y<8?
４２ y←0
４３ x←0
４４ dd←0
４５ u←0
４６ dd←dd+t(y,x)×cos((2x+1)uπ/16)
４７ u←u+1
４８ u<8?
４９ Buffer(y,u)←dd
５０ x←x+1
５１ x<8?
５２ y←y+1
５３ y<8?
RET

【図１２】


逆ＤＣＴ２
６０ y←0
６１ v←0
６２ dd←0
６３ u←0
６４ dd←dd+Buffer(y,u)×cos((2x+1)uπ/16)
６５ u←u+1
６６ u<umax(y)
６７ t(v,y)←dd
６８ v←v+1
６９ v<8?
７０ y←y+1
７１ y<8?
７２ y←0
７３ x←0
７４ dd←0
７５ u←0
７６ dd←dd+t(y,u)×cos((2x+1)uπ/16)
７７ u←u+1
７８ u<8?
７９ Buffer(x,y)←dd
８０ x←x+1
umax(y)={5 (y=1,2)
4 (y=3)
0 (y=0)
８１ x<8?
８２ y←y+1
８３ y<8?
RET

【図１５】


VLD
90
符号を8bit 取り出す
91
8bit の値から符号の種類を判断する
92
8bit以下の符号？
Y
N
93
さらに必要なｂｉｔ数分取り出す
94
符号のbit 分符号バッファのポインタを加算する
95
符号に対応するラン長とレベルを取り出す
96
ラン長とレベルを返す
RET
(A)
VLD
97
符号を20 bit取り出す
98
bit0～bit7からレベルを取り出す
99
bit8～bit13 からラン長を取り出す
100
符号のバッファのポインタを20bit 分加算する
104
ラン長とレベルを返す
RET
(B)